

05/2005

ISR④

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-299081

(43)Date of publication of application : 17.10.2003

(51)Int.Cl.

H04N 7/24

(21)Application number : 2002-104315

(71)Applicant : SONY CORP

(22)Date of filing : 05.04.2002

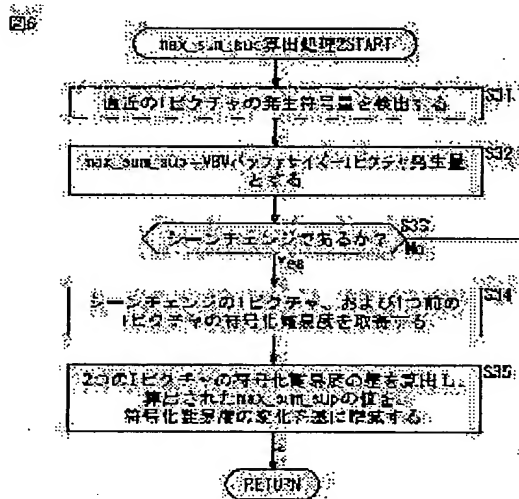
(72)Inventor : UENO HIROMICHI

(54) ENCODER, ENCODING METHOD, PROGRAM AND RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To decide the maximum value of a sum value sum-sup of a bit supply quantity, according to the margin of a VBV (video buffering verifier) buffer size.

SOLUTION: In a step S31, the generating code amount of a nearest I picture is detected, and in a step S32, it is determined that $\text{max_sum_sup} = \text{VBV buffer size} - \text{I picture generating amount}$. In a step S33, it is judged that a scene change occurs. If it is judged that the scene change occurs in a step S34, the encoding difficulties of the I picture of the scene change and an I picture before one picture are obtained. In a step S35, the difference between the two I pictures in coding difficulty are calculated, and the value of max-sum-sup calculated in the step S32 is increased or decreased on the basis of the difference in the coding difficulty, i.e., a scene change from a difficult image to an easy image or a scene change from the easy image to the difficult image.



LEGAL STATUS

[Date of request for examination]

07.03.2005

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号
特開2003-299081
(P2003-299081A)

(43)公開日 平成15年10月17日(2003. 10. 17)

(51)Int.Cl.⁷
H 0 4 N 7/24

識別記号

F I
H 0 4 N 7/13

テーマコード(参考)
Z 5 C 0 5 9

審査請求 未請求 請求項の数5 O L (全 11 頁)

(21)出願番号 特願2002-104315(P2002-104315)

(22)出願日 平成14年4月5日(2002. 4. 5)

(71)出願人 000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号
(72)発明者 上野 弘道
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(74)代理人 100082131
弁理士 稲本 義雄

最終頁に続く

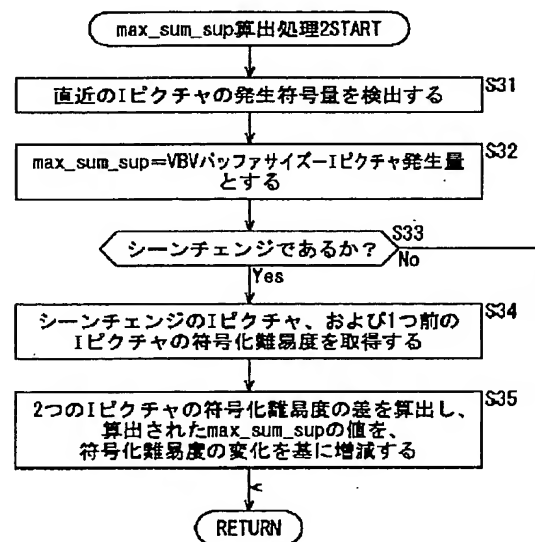
(54)【発明の名称】 符号化装置および符号化方法、プログラム、並びに記録媒体

(57)【要約】

【課題】 ビット補給量supplementの積算値sum_supの最大値を、VBVバッファサイズの余裕度に応じて決定する。

【解決手段】 ステップS31で、直近のIピクチャの発生符号量を検出され、ステップS32で、max_sum_sup=VBVバッファサイズ-Iピクチャ発生量とされる。ステップS33で、シーンチェンジであるか否かが判断され、シーンチェンジであると判断された場合、ステップS34で、シーンチェンジのIピクチャ、および1つ前のIピクチャの符号化難易度が取得されて、ステップS35で、2つのIピクチャの符号化難易度の差が算出されて、ステップS32で算出されたmax_sum_supの値が、符号化難易度の差、すなわち、難しい絵柄から簡単な絵柄へのシーンチェンジであるか、簡単な絵柄から難しい絵柄へのシーンチェンジであるかを基に増減される。

図6



【特許請求の範囲】

【請求項1】 非圧縮データの符号化を行う符号化装置において、

前記非圧縮データの複雑さを検出する第1の検出手段と、

前記非圧縮データを圧縮符号化する符号化手段と、

前記第1の検出手段により検出された、前記符号化手段により過去に符号化された前記非圧縮データの複雑さを基に、これから符号化される前記非圧縮データに対して割り当てられるVBVバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量を算出する算出手段と、

前記符号化手段により過去に符号化された前記非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出する第2の検出手段と、

前記VBVバッファのバッファ容量から、前記第2の検出手段により検出された前記フレーム内符号化画像のビット発生量を減算した値を算出し、前記第2のビット量の合計値の最大値として設定する設定手段とを備えることを特徴とする符号化装置。

【請求項2】 1つ前のピクチャと次に符号化処理するピクチャとで、絵柄が所定の基準値より変化したことを検出する第3の検出手段と、

前記第3の検出手段により、前記絵柄が所定の前記基準値より変化したことが検出された場合、前記第1の検出手段により検出された絵柄の変化前後の前記フレーム内符号化画像の複雑さに基づいて、前記設定手段により設定された前記第2のビット量の合計値の最大値を再設定する再設定手段とを更に備えることを特徴とする請求項1に記載の符号化装置。

【請求項3】 非圧縮データの符号化を行う符号化装置の符号化方法において、

前記非圧縮データの複雑さを検出する第1の検出ステップと、

前記非圧縮データを圧縮符号化する符号化ステップと、
前記第1の検出ステップの処理により検出された、前記符号化ステップの処理により過去に符号化された前記非圧縮データの複雑さを基に、これから符号化される前記非圧縮データに対して割り当てられるVBVバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量を算出する算出ステップと、
前記符号化ステップの処理により過去に符号化された前記非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出する第2の検出ステップと、

前記VBVバッファのバッファ容量から、前記第2の検出ステップの処理により検出されたフレーム内符号化画像のビット発生量を減算した値を算出し、前記第2のビット量の合計値の最大値に設定する設定ステップとを含むことを特徴とする符号化方法。

【請求項4】 非圧縮データの符号化を行う符号化装置

用のプログラムであって、

前記非圧縮データの複雑さを検出する第1の検出ステップと、

前記非圧縮データを圧縮符号化する符号化ステップと、
前記第1の検出ステップの処理により検出された、前記符号化ステップの処理により過去に符号化された前記非圧縮データの複雑さを基に、これから符号化される前記非圧縮データに対して割り当てられるVBVバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量を算出する算出ステップと、
前記符号化ステップの処理により過去に符号化された前記非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出する第2の検出ステップと、

前記VBVバッファのバッファ容量から、前記第2の検出ステップの処理により検出されたフレーム内符号化画像のビット発生量を減算した値を算出し、前記第2のビット量の合計値の最大値に設定する設定ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項5】 非圧縮データの符号化を行う符号化装置を制御するコンピュータが実行可能なプログラムであって、

前記非圧縮データの複雑さを検出する第1の検出ステップと、

前記非圧縮データを圧縮符号化する符号化ステップと、
前記第1の検出ステップの処理により検出された、前記符号化ステップの処理により過去に符号化された前記非圧縮データの複雑さを基に、これから符号化される前記非圧縮データに対して割り当てられるVBVバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量を算出する算出ステップと、
前記符号化ステップの処理により過去に符号化された前記非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出する第2の検出ステップと、

前記VBVバッファのバッファ容量から、前記第2の検出ステップの処理により検出されたフレーム内符号化画像のビット発生量を減算した値を算出し、前記第2のビット量の合計値の最大値に設定する設定ステップとを含むことを特徴とするプログラム。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、符号化装置および符号化方法、プログラム、並びに記録媒体に関し、特に、フィードバック型レート制御において、ビット補給レート制御を行う場合に用いて好適な、符号化装置および符号化方法、プログラム、並びに記録媒体に関する。

【0002】

【従来の技術】 近年、映像データおよび音声データを圧縮して情報量を減らす方法として、種々の圧縮符号化方法が提案されており、その代表的なものにMP E G 2

(MovingPicture Experts Group Phase 2) がある。

【0003】このような画像圧縮方式において、良好なエンコード画質を得る方法として、TM5 (Test Model 5) がある。TM5のステップ1においては、ピクチャ単位に与えるターゲットビットの算出を行う。ターゲットビットの算出においては、ピクチャタイプ別のGC (Global Complexity) のそれぞれの比率に応じて、そのGOP (Group of Picture) 内の残りのピクチャに割り当てることができるビット量Rを比例配分して、各ピクチャに割り当てられるビット量を算出する。

【0004】TM5は、GOPあたりの発生ビット量をほぼ一定にするために優れた方法であるが、固定レート符号化を行う場合には、必ずしも、GOPの発生ビット量を一定にする必要はない。固定レート符号化においては、VBV (Video Buffering Verifier) バッファの占有量が、規定値をオーバーフロー、あるいはアンダーフローしないようにしなければならない。

【0005】TM5においては、GOPあたりの発生ビット量がほぼ一定であるから、VBVバッファがオーバーフローあるいはアンダーフローすることはない。しかしながら、TM5においては、低いビットレートで符号化した場合に、バッファ容量を有効利用することができない。例えば、MPEGのMP@PLにおいて、TM5を適用した場合、VBVバッファ容量は約1.8Mbitであるのに対して、バッファから引き抜かれる1枚あたりのピクチャのビット量が少ないため、約1.8Mbitを有効に利用することができない。

【0006】このように、入力される絵柄に関わらず、一定量のビット量を割り当ててしまうことにより、符号化難易度が高い絵柄については、符号化歪みが顕著に発生してしまい、一方、符号化難易度が低い絵柄は、符号化歪みが少ないため、全体として、むらの多い不安定な画像になってしまう。

【0007】このような問題を解決するために、符号化難易度が高い絵柄には、バッファがアンダーフローしない範囲で、より多くのビット量を配分し、一方、符号化難易度が低い絵柄には、バッファがオーバーフローしない範囲で、絵柄に適した少ないビット量を配分する必要がある。

【0008】そこで、本出願人は、特開平10-75443において、映像データの部分毎の絵柄の複雑さに応じて発生ビット量を調節し、全体として、圧縮後の映像の品質を向上させることができるようにした、映像データ圧縮装置およびその方法について開示している。

【0009】TM5において、GOPの残りのピクチャに割り当てることができる使用可能ビット量Rは、レートコントロールで重要なパラメータである。例えば、GOPの前半において、複雑な絵柄の画像が続いたために、たくさんのビット量を割り当ててしまうと、GOPの後半で、ビット量Rが、極端に少なくなってしまう

り、あるいは、負の数になってしまう。

【0010】これに対して、本出願人が特開平10-75443において開示したビット補給レート制御とは、これからエンコードしようとする複数枚のピクチャに対して割り当てられている使用可能ビット量Rに、そのエンコード対象の画像難易度やVBVバッファ占有量に応じて、ビット量を加える、あるいは減じる（以下、加えられる、あるいは減じられるビットをsupplementと称する）ことを特徴とするレート制御方式である。

【0011】

【発明が解決しようとする課題】以前提案されたビット補給レート制御は、これからエンコードしようとする複数枚のピクチャ画像難易度等の情報が全て既知である場合、すなわちエンコード情報を先読みしたフィードフォワード (Feed Forward) 型レート制御に適用されていたもので、例えば、GOPの15枚のデータを蓄積した後、その画像符号化難易度を判断していたので、その情報蓄積に一定の遅延を生じてしまうものである。

【0012】しかしながら、先読み情報を得ることができないフィードバック (Feed Back) 型レート制御では、未来のVBV余裕度を正確に見積もることができないため、sum_supplement (以下、sum_supと称する) の最大値および最小値をビットレートや使用可能VBVサイズによって決定した固定値を用いざるを得なかった。しかしながら、特に、sum_supの最大値が固定値の場合、ピクチャの発生量次第によってはVBVアンダーフローを起こしやすくなるなどの問題があり、VBVアンダーフローを起こさないようにするためには、VBV余裕度に応じて、sum_supの最大値を決定する必要がある。

【0013】本発明はこのような状況に鑑みてなされたものであり、フィードバック型レート制御において、ビット補給レート制御を行うことができるようにするものである。

【0014】

【課題を解決するための手段】本発明の符号化装置は、非圧縮データの複雑さを検出する第1の検出手段と、非圧縮データを圧縮符号化する符号化手段と、第1の検出手段により検出された、符号化手段により過去に符号化された非圧縮データの複雑さを基に、これから符号化される非圧縮データに対して割り当てられるVBVバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量を算出する算出手段と、符号化手段により過去に符号化された非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出する第2の検出手段と、VBVバッファのバッファ容量から、第2の検出手段により検出されたフレーム内符号化画像のビット発生量を減算した値を算出し、第2のビット量の合計値の最大値として設定する設定手段とを備えることを特徴とする。

【0015】1つ前のピクチャと次に符号化処理するピクチャとで、絵柄が所定の基準値より変化したことを検出する第3の検出手段と、第3の検出手段により、絵柄が所定の基準値より変化したことが検出された場合、第1の検出手段により検出された絵柄の変化前後のフレーム内符号化画像の複雑さに基づいて、設定手段により設定された第2のビット量の合計値の最大値を再設定する再設定手段とを更に備えさせるようにすることができる。

【0016】本発明の符号化方法は、非圧縮データの複雑さを検出する第1の検出ステップと、非圧縮データを圧縮符号化する符号化ステップと、第1の検出ステップの処理により検出された、符号化ステップの処理により過去に符号化された非圧縮データの複雑さを基に、これから符号化される非圧縮データに対して割り当てられるVＢＶバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量を算出する算出ステップと、符号化ステップの処理により過去に符号化された非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出する第2の検出ステップと、VＢＶバッファのバッファ容量から、第2の検出ステップの処理により検出されたフレーム内符号化画像のビット発生量を減算した値を算出し、第2のビット量の合計値の最大値に設定する設定ステップとを含むことを特徴とする。

【0017】本発明の記録媒体に記録されているプログラムは、非圧縮データの複雑さを検出する第1の検出ステップと、非圧縮データを圧縮符号化する符号化ステップと、第1の検出ステップの処理により検出された、符号化ステップの処理により過去に符号化された非圧縮データの複雑さを基に、これから符号化される非圧縮データに対して割り当てられるVＢＶバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量を算出する算出ステップと、符号化ステップの処理により過去に符号化された非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出する第2の検出ステップと、VＢＶバッファのバッファ容量から、第2の検出ステップの処理により検出されたフレーム内符号化画像のビット発生量を減算した値を算出し、第2のビット量の合計値の最大値に設定する設定ステップとを含むことを特徴とする。

【0018】本発明のプログラムは、非圧縮データの複雑さを検出する第1の検出ステップと、非圧縮データを圧縮符号化する符号化ステップと、第1の検出ステップの処理により検出された、符号化ステップの処理により過去に符号化された非圧縮データの複雑さを基に、これから符号化される非圧縮データに対して割り当てられるVＢＶバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量を算出する算出ステップと、符号化ステップの処理により過去に符

号化された非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出する第2の検出ステップと、VＢＶバッファのバッファ容量から、第2の検出ステップの処理により検出されたフレーム内符号化画像のビット発生量を減算した値を算出し、第2のビット量の合計値の最大値に設定する設定ステップとを含むことを特徴とする。

【0019】本発明の符号化装置および符号化方法、並びにプログラムにおいては、非圧縮データの複雑さが検出され、非圧縮データが圧縮符号化され、過去に符号化された非圧縮データの複雑さを基に、これから符号化される非圧縮データに対して割り当てられるVＢＶバッファのバッファ容量のうち、使用可能である第1のビット量に加えられる第2のビット量が算出され、符号化ステップの処理により過去に符号化された非圧縮データのうちの、フレーム内符号化画像のビット発生量を検出され、VＢＶバッファのバッファ容量から、検出されたフレーム内符号化画像のビット発生量を減算した値が算出されて、第2のビット量の合計値の最大値に設定される。

【0020】

【発明の実施の形態】以下、図を参照して、本発明の実施の形態について説明する。

【0021】図1は、本発明を適応したエンコーダ1の構成を示すブロック図である。

【0022】画像並び替え部12は、入力された非圧縮映像データを符号化順に並べ替える。走査変換・マクロブロック化部13は、ピクチャ・フィールド変換を行い、例えば、非圧縮映像データが映画の映像データである場合、3：2プルダウン処理等を行う。イントラＡＣ算出部14は、画像並び替え部12および走査変換・マクロブロック化部13により処理され、1ピクチャに圧縮符号化されるピクチャから、イントラＡＣ（intra A C）を算出する。

【0023】1ピクチャについては、他のピクチャの参照なしに圧縮符号化されるため、後述するME残差を求めることができない。従って、1ピクチャの符号化難易度を求めるために、ME残差に代わるパラメータとして、イントラＡＣが用いられる。イントラＡＣは、MP E G方式におけるD C T処理単位のD C Tブロックごとの映像データとの分散値の総和として定義されるパラメータであって、映像の複雑さを指標し、映像の絵柄の難しさおよび圧縮後のデータ量と相関性を有する。すなわち、イントラＡＣとは、D C Tブロック単位で、それぞれの画素の画素値から、ブロック毎の画素値の平均値を引いたものの絶対値和の、画面内における総和である。イントラＡＣは、次の式（1）で示される。

【0024】

【数1】

$$\text{IntraAC} = \sum_{\text{picture}} \sum_{k=1}^{64} |P_k - \bar{P}|$$

・・・ (1)

【0025】また、式 (1) において、式 (2) が成り立つ。

【数2】

$$\bar{P} = \frac{1}{64} \sum_{k=1}^{64} P_k$$

・・・ (2)

【0026】イントラAC算出部14は、算出されたイントラACの値を、レートコントロール部15の難易度算出部32に出力する。

【0027】演算処理部16は、動き補償部25から供給される動き補償情報を基に、供給された映像データに対して動き補償を行い、DCT部18に対して出力する。DCT部18は、演算処理部16から入力された映像データに対して、例えば、16画素×16画素のマクロブロック単位に離散コサイン変換(DCT)処理を施し、時間領域のデータから周波数領域のデータに変換して、量子化部19に対して出力する。

【0028】量子化部19は、DCT部18から入力された周波数領域のデータを、レートコントロール部15の量子化インデックス決定部35から供給される量子化インデックスQで量子化し、量子化データとしてVLC(Variable Length Code; 可変長符号化)部20および逆量子化部22に対して出力する。

【0029】VLC部20は、量子化部19から入力された量子化データに対し、所定の変換テーブルに基づく可変長符号化処理を行い、その結果得られる可変長符号化データをバッファ21に出力する。

【0030】バッファ21は、入力された符号化データをバッファリングし、符号化ビットストリームとして、順次、出力する。

【0031】逆量子化部22は、量子化部19から入力された量子化データを、量子化部19が実行した量子化の量子化ステップで逆量子化し、逆量子化データとして逆DCT部23に対して出力する。

【0032】逆DCT部23は、逆量子化部22から入力される逆量子化データに対して逆DCT処理を行い、演算処理部24に対して出力する。

【0033】演算処理部24は、動き補償部25の出力データ、および逆DCT部23の出力データを加算し、動き補償部25に対して出力する。動き検出部17は、圧縮対象となるピクチャ(入力ピクチャ)の注目マクロブロックと、参照されるピクチャ(参照ピクチャ)との間の差分値の絶対値和あるいは自乗値和が最小となるようなマクロブロックを探し、動きベクトルを求めて、動き補償部25に出力する。動き補償部25は、演算処理部24の出力データに対して、動き検出部17から入力

される動きベクトルに基づいて動き補償処理を行い、演算処理部24、および演算処理部16に対して出力する。

【0034】レートコントロール部15は、ME残差算出部31、難易度算出部32、genbit検出部33、ターゲットビット決定部34、および量子化インデックス決定部35で構成され、ターゲットビットおよび量子化インデックスを決定する。

【0035】ME残差算出部31は、画像の符号化難易度と強い相関があるパラメータであるME残差を算出する。動き予測によって、参照フレームから入力フレームへの差分値の絶対値和などが少なくなるような動きベクトルを求めることができるが、その場合における差分値の絶対値和、あるいは自乗和などで求められる誤差成分のパワーがME残差である。Pピクチャ、およびBピクチャにおいては、ME残差と画像の符号化難易度とは、ほぼ単純な比例関係を有している。

【0036】難易度算出部32は、ME残差算出部31から入力されるME残差による近似により、式(3)、および、式(4)を用いて、PピクチャおよびBピクチャの符号化難易度Djを算出する。

【数3】

$$D_j = a_p \times ME_j \times b_p$$

・・・ (3)

【数4】

$$D_j = a_B \times ME_j \times b_B$$

・・・ (4)

【0037】ここで、MEjは、j番目のピクチャにおけるME残差であり、ap、aB、bp、bBは、それぞれ、1次式で近似した場合の傾きと補正值である。

【0038】また、難易度算出部32はイントラAC算出部14から入力されるイントラACによる近似により、同様にIピクチャの符号化難易度Djを算出し、ターゲットビット決定部34に出力する。

【0039】そして、難易度算出部32は、それぞれのピクチャで算出された符号化難易度Djから、GOP毎の難易度平均avgDを算出する。

【0040】genbit検出部33は、バッファ21にバッファリングされている符号化データから、直前に符号化されたIピクチャの発生ビット量genbitを検出し、その値を、ターゲットビット決定部34に出力する。

【0041】ターゲットビット決定部34は、難易度算出部32から入力された符号化難易度Dj、および、genbit検出部33から入力されたIピクチャの発生ビット量genbitに基づいて、各ピクチャタイプのピクチャそれぞれのターゲットビットを算出して、レート制御を行う。

【0042】すなわち、ターゲットビット決定部34は、後述する処理により、エンコードを終了した過去の

画像における難易度などを基に、これからエンコードしようとする複数枚のピクチャに対して割り当てられている使用可能ビット量 R に加えられるsupplementの値(supplementは、正の値である場合、負の値である場合、0である場合がある)を決定する。ターゲットビット決定部34は、この使用可能ビット量 R +supplementを基に、ターゲットビットの値を求め、量子化インデックス決定部35に出力する。

【0043】量子化インデックス決定部35は、ターゲットビット決定部34から入力されたターゲットビットの値に基づいて、量子化インデックス Q を生成し、量子化部19に対して出力する。

【0044】次に、図2のフローチャートを参照して、エンコードを終了した過去の画像における難易度を基に R に加えるsupplementを決定する、ビット補給レート制御処理について説明する。

【0045】ステップS1において、ターゲットビット決定部34は、現在処理中のピクチャは、GOPの先頭であるか否かを判断する。ステップS1において、GOPの先頭ではないと判断された場合、GOPの先頭であると判断されるまで、ステップS1の処理が繰り返される。

【0046】ステップS1において、GOPの先頭であると判断された場合、ステップS2において、ターゲットビット決定部34は、難易度算出部32より、前のGOPにおける難易度平均 $avgD$ を取得する。

【0047】ステップS3において、図3、もしくは図6を用いて後述する max_sum_sup 算出処理が実行される。

【0048】ステップS4において、ターゲットビット決定部34は、 $avgD > 0x2000$ かつ $sum_sup < max_sum_sup$ であるか否かを判断する。ここで、難易度平均 $avgD$ と比較されている $0x2000$ は、予め定められた閾値であり、画質を検討しながら要求される画質を得るために設定可能な値である。

【0049】ステップS4において、 $avgD > 0x2000$ かつ $sum_sup < max_sum_sup$ であると判断された場合、ステップS5において、ターゲットビット決定部34は、使用可能ビット量 R に対して、正の値のsupplementを加える。すなわち、ターゲットビット決定部34は、前のGOPは、ある一定以上の難易度を有していたため、これからエンコードするGOPの難易度を、前のGOPと同程度であると予測して、使用可能ビット量 R に対して、正の値のsupplementを加える。

【0050】ステップS4において、 $avgD > 0x2000$ かつ $sum_sup < max_sum_sup$ ではないと判断された場合、ステップS6において、ターゲットビット決定部34は、 $avgD < 0x1000$ 、かつ $sum_sup > min_sum_sup$ であるか否かを判断する。ここで、難易度平均 $avgD$ と比較されている $0x1000$ は、予め定められた閾値であり、上述し

た $0x2000$ より小さな値(画像難易度が低いことを示す値)であり、画質を検討しながら要求される画質を得るために設定可能な値である。

【0051】ステップS6において、 $avgD < 0x1000$ 、かつ $sum_sup > min_sum_sup$ であると判断された場合、ステップS7において、ターゲットビット決定部34は、使用可能ビット量 R に対して、負の値のsupplementを加える。すなわち、ターゲットビット決定部34は、前のGOPは、ある一定以下の難易度であった(すなわち、簡単な画像であった)ため、これからエンコードするGOPの難易度を、前のGOPと同程度であると予測して、使用可能ビット量 R に対して、負の値のsupplementを加える。

【0052】ステップS6において、 $avgD < 0x1000$ 、かつ $sum_sup > min_sum_sup$ ではなかったと判断された場合、ステップS8において、ターゲットビット決定部34は、 $supplement = 0$ とする。すなわち、ターゲットビット決定部34は、使用可能ビット量 R に対して、supplementの増減を行わない。

【0053】ステップS5、ステップS7、もしくはステップS8の処理の終了後、ステップS9において、ターゲットビット決定部34は、ステップS5、ステップS7、もしくはステップS8の処理において用いられたsupplementの値を用いて、 $sum_sup = sum_sup + supplement$ とし、処理は、ステップS1に戻り、それ以降の処理が繰り返される。

【0054】図2を用いて説明した処理により、エンコードを終了した過去の画像における難易度を基に、使用可能ビット量 R に加える、あるいは、減少されるsupplementの値が決定される。例えば、GOP単位で、 $R + supplement$ (supplementは、正の値であるか、負の値であるか、もしくは0である)が決定される場合、前のGOPの画像難易度(イントラAC、あるいは、ME残差等)の平均値を基に、これからエンコードするGOPの難易度が前のGOPの難易度と同程度であると予測して、使用可能ビット量 R に対して、その難易度に応じたsupplementが加えられる。

【0055】ここでは、画像難易度をイントラAC、あるいは、ME残差を用いて算出するものとして説明したが、画像難易度は、それ以外のパラメータを用いて算出するようにしても良い。

【0056】また、supplementの具体的な値の算出方法は、例えば、特開平10-75443に開示されている方法でも良いし、それ以外の方法で、要求される画質を得ることができるsupplementの値を用いるようにしても良い。

【0057】また、ここでは、前の1GOPにおける難易度平均 $avgD$ を用いるものとして説明したが、難易度算出部32は、1GOPにおける難易度平均 $avgD$ に代わって、例えば、複数のGOP、もしくは、GOPの一

部における難易度平均を求めるようにしても良いし、更に、単純な難易度平均ではなく、必要に応じて、重み付け和や重み付け平均を算出するようにしても良い。

【0058】次に、図2のステップS3において実行されるmax_sum_sup算出処理について説明する。

【0059】LongGOPにおいては、1ピクチャの発生量が大きくなる傾向がある。従って、ピクチャの発生量によってVBVアンダーフローを起こすことを防ぐためには、VBVバッファサイズからエンコードを終了した直近の1ピクチャのビット発生量を引いたものをsum_supの最大値(max_sum_sup)とすればよい。

【0060】図3のフローチャートを参照して、図2のステップS3において実行されるmax_sum_sup算出処理1について説明する。

【0061】ステップS21において、genbit検出部33は、直近の1ピクチャの発生符号量genbitを検出する。ターゲットビット決定部34は、genbit検出部33から、genbitの値の入力を受ける。

【0062】ステップS22において、ターゲットビット決定部34は、sum_supの最大値であるmax_sum_supの値を、 $\text{max_sum_sup} = \text{VBVバッファサイズ} - 1 \text{ピクチャ発生量}$ とし、処理は、図2のステップS4に戻る。

【0063】図3を用いて説明した処理により、図4に示すように、VBVサイズから1ピクチャ発生量を引いた、実線矢印の合計量が、VBV余裕度として、次のGOPのsum_supの最大値とされる。これにより、アンダーフローしやすい絵柄ではsupplementが与えられにくくなり、アンダーフローに対する余裕がある絵柄に対してはsupplementが与えられやすくなる。すなわち、1ピクチャのビット発生量が多いために発生するVBVアンダーフローを防ぐことができる。

【0064】しかしながら、図3を用いて説明した処理では、シーンチェンジが起きた場合に不具合が発生してしまう。例えば、難しい絵柄から簡単な絵柄へのシーンチェンジが起きた場合、図5に示されるように、前のGOPが難しい絵柄のため、次のGOPのmax_sum_sup(実線矢印の合計)が大きくなり、絵柄が簡単なGOPに、大きくなったmax_sum_supを適用してしまうので、VBVの余裕が無いものに対してsum_supの最大値を大きくしてしまう。また、同様に、簡単な絵柄から難しい絵柄へのシーンチェンジにおいても、逆の不具合が発生してしまう。

【0065】これを防ぐために、シーンチェンジの1ピクチャを含むGOPをエンコードする場合には、前の1ピクチャ発生量により求められたsum_supの最大値を、シーンチェンジの1ピクチャの難易度により増減させるようにすることができる。

【0066】図6のフローチャートを参照して、図2のステップS3において実行されるmax_sum_sup算出処理2について説明する。

【0067】ステップS31において、genbit検出部33は、直近の1ピクチャの発生符号量genbitを検出する。ターゲットビット決定部34は、genbit検出部33から、genbitの値の入力を受ける。

【0068】ステップS32において、ターゲットビット決定部34は、sum_supの最大値であるmax_sum_supの値を、 $\text{max_sum_sup} = \text{VBVバッファサイズ} - 1 \text{ピクチャ発生量}$ とする。

【0069】ステップS33において、ターゲットビット決定部34は、シーンチェンジであるか否かを判断する。シーンチェンジであるか否かの判断は、例えば、ME残差算出部31により算出されるME残差の値を基にして判断するようにしても良いし、それ以外のいかなる方法によって判断するようにしても良い。

【0070】ステップS33において、シーンチェンジではないと判断された場合、処理は、図2のステップS4に戻る。

【0071】ステップS33において、シーンチェンジであると判断された場合、ステップS34において、ターゲットビット決定部34は、難易度算出部32より、シーンチェンジの1ピクチャ、および1つ前の1ピクチャの符号化難易度を取得する。

【0072】ステップS35において、ターゲットビット決定部34は、2つの1ピクチャの符号化難易度の差を算出し、ステップS32において算出されたmax_sum_supの値を、符号化難易度の差、すなわち、難しい絵柄から簡単な絵柄へのシーンチェンジであるか、簡単な絵柄から難しい絵柄へのシーンチェンジであるかを基に増減して、処理は、図2のステップS4に戻る。

【0073】具体的には、シーンチェンジ後の符号化難易度が低い場合は、max_sum_supの値を少なくし、シーンチェンジ後の符号化難易度が高い場合は、max_sum_supの値を多くする。

【0074】図6を用いて説明した処理により、シーンチェンジの1ピクチャを含むGOPをエンコードする場合には、前の1ピクチャ発生量により求めたsum_supの最大値をシーンチェンジの1ピクチャの難易度により増減させることにより、例えば、次のGOPの1ピクチャ発生量が大きく、VBVに余裕が無いにもかかわらず、大きなsum_sup最大値となってしまうようなことをふせぐようにすることができる。

【0075】また、本発明は、図2を用いて説明したビット補給レート制御処理以外でも、ビット補給レート制御を行う場合、すなわち、ビット補給量supplementの積算値sum_supの最大値である、max_sum_supを用いる処理の全てに適用可能である。

【0076】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。この場合、例えば、エンコーダ1は、図7に示されるようなパーソナルコンピュータ10

1により構成される。

【0077】図7において、CPU111は、ROM112に記憶されているプログラム、または記憶部118からRAM113にロードされたプログラムに従って、各種の処理を実行する。RAM113にはまた、CPU111が各種の処理を実行する上において必要なデータなども適宜記憶される。

【0078】CPU111、ROM112、およびRAM113は、バス114を介して相互に接続されている。このバス114にはまた、入出力インタフェース115も接続されている。

【0079】入出力インタフェース115には、キーボード、マウスなどよりなる入力部116、ディスプレイやスピーカなどよりなる出力部117、ハードディスクなどより構成される記憶部118、モデム、ターミナルアダプタなどより構成される通信部119が接続されている。通信部119は、インターネットを含むネットワークを介しての通信処理を行う。

【0080】入出力インタフェース115にはまた、必要に応じてドライブ120が接続され、磁気ディスク131、光ディスク132、光磁気ディスク133、あるいは、半導体メモリ134などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部118にインストールされる。

【0081】一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、ネットワークや記録媒体からインストールされる。

【0082】この記録媒体は、図7に示されるように、装置本体とは別に、ユーザにプログラムを供給するために配布される、プログラムが記憶されている磁気ディスク131（フロッピディスクを含む）、光ディスク132（CD-ROM（Compact Disk-Read Only Memory）、DVD（Digital Versatile Disk）を含む）、光磁気ディスク133（MD（Mini-Disk）（商標）を含む）、もしくは半導体メモリ134などよりなるパッケージメディアにより構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに供給される、プログラムが記憶されているROM112や、記憶部118に含まれるハードディスクなどで構成される。

【0083】なお、本明細書において、記録媒体に記憶されるプログラムを記述するステップは、含む順序に沿

って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0084】

【発明の効果】本発明によれば、画像データをエンコードすることができる。また、本発明によれば、エンコードを終了した過去の画像における難易度を基に使用可能ビット量Rに加えるsupplementを決定する場合の、supplementの合計値の最大値を設定することができるので、フィードバック型レート制御にビット補給レート制御を適用する場合にVBVアンダーフローを防ぐことができる。

【0085】また、シーンチェンジが起きたGOPをエンコードする際には、シーンチェンジ前後のフレーム内符号化画像の画像難易度を比較した値を用いて、supplementの合計値の最大値を再設定することができるので、フィードバック型レート制御にビット補給レート制御を適用する場合にVBVアンダーフローを防ぐことができる。

【図面の簡単な説明】

【図1】本発明を適用したエンコーダの構成を示すブロック図である。

【図2】ビット補給レート制御処理について説明するフローチャートである。

【図3】max_sum_sup算出処理1について説明するフローチャートである。

【図4】VBVバッファと、sum_supの最大値とについて説明するための図である。

【図5】VBVバッファと、sum_supの最大値とについて説明するための図である。

【図6】max_sum_sup算出処理2について説明するフローチャートである。

【図7】パーソナルコンピュータの構成について説明する図である。

【符号の説明】

1 エンコーダ、 12 画像並び替え部、 13 走査変換・マクロブロック化部、 14 イントラAC算出部、 15 レートコントロール部、 16 演算処理部、 17 動き検出部、 18 DCT処理部、 19 量子化部、 20 VLC部、 21 バッファ、 22 逆量子化部、 23 逆DCT処理部、 24 演算処理部、 25 動き補償部、 31 ME残差算出部、 32 難易度算出部、 33 genbit検出部、 34 ターゲットビット決定部、 35 量子化インデックス決定部

[illegible]

図2

```
graph TD
    Start([ビット補給レート制御処理start]) --> S1{S1  
GOPの先頭であるか?}
    S1 -- Yes --> S2[S2  
前のGOPにおける  
難易度平均avgDを取得する]
    S1 -- No --> S4
    S2 --> S3[S3  
max_sum_sup算出処理]
    S3 --> S4{S4  
avgD > 0x2000かつ  
sum_sup < max_sum_sup  
であるか?}
    S4 -- Yes --> S5[S5  
R+supplement(>0)とする]
    S4 -- No --> S6{S6  
avgD < 0x1000かつ  
sum_sup > min_sum_sup  
であるか?}
    S5 --> S9[S9  
sum_sup = sum_sup + supplementとする]
    S6 -- Yes --> S7[S7  
R+supplement(<0)とする]
    S6 -- No --> S8[S8  
supplement = 0とする]
    S7 --> S9
    S8 --> S9
```

Flowchart illustrating the Bit Rate Control Processing (ビット補給レート制御処理start):

- S1**: Decision: GOPの先頭であるか? (Is it the start of a GOP?).
 - If **Yes**, proceed to **S2**.
 - If **No**, proceed to **S4**.
- S2**: Process: 前のGOPにおける難易度平均avgDを取得する (Obtain the average difficulty avgD of the previous GOP).
- S3**: Process: max_sum_sup算出処理 (max_sum_sup calculation processing).
- S4**: Decision: avgD > 0x2000かつ sum_sup < max_sum_sup であるか? (Is avgD > 0x2000 and sum_sup < max_sum_sup?).
 - If **Yes**, proceed to **S5**.
 - If **No**, proceed to **S6**.
- S5**: Process: R+supplement(>0)とする (Set R+supplement(>0)).
- S6**: Decision: avgD < 0x1000かつ sum_sup > min_sum_sup であるか? (Is avgD < 0x1000 and sum_sup > min_sum_sup?).
 - If **Yes**, proceed to **S7**.
 - If **No**, proceed to **S8**.
- S7**: Process: R+supplement(<0)とする (Set R+supplement(<0)).
- S8**: Process: supplement = 0とする (Set supplement = 0).
- S9**: Process: sum_sup = sum_sup + supplementとする (Set sum_sup = sum_sup + supplement).

図3

```
graph TD; A([max_sum_sup算出処理1START]) --> B[直近のIピクチャの発生符号量を検出する S21]; B --> C["max_sum_sup=VBVバッファサイズ-Iピクチャ発生量とする S22"]; C --> D([RETURN]);
```

The flowchart illustrates the process for calculating max_sum_sup. It begins with an oval labeled "max_sum_sup算出処理1START". An arrow points down to a rectangular process block labeled "直近のIピクチャの発生符号量を検出する" with the label "S21" to its right. Another arrow points down to a second rectangular process block labeled "max_sum_sup=VBVバッファサイズ-Iピクチャ発生量とする" with the label "S22" to its right. A final arrow points down to an oval labeled "RETURN".

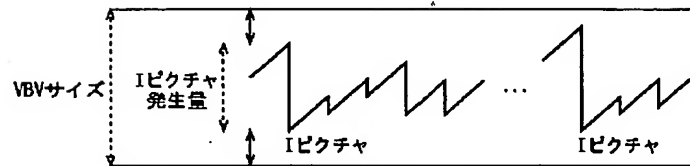
```

graph TD
    Start([max_sum_sup算出処理2START]) --> S31[直近のIピクチャの発生符号量を検出する]
    S31 --> S32["max_sum_sup = VBVバッファサイズ - Iピクチャ発生量とする"]
    S32 --> S33{シーンチェンジであるか?}
    S33 -- No --> S33
    S33 -- Yes --> S34[シーンチェンジのIピクチャ、および1つ前のIピクチャの符号化難易度を取得する]
    S34 --> S35[2つのIピクチャの符号化難易度の差を算出し、算出されたmax_sum_supの値を、符号化難易度の変化を基に増減する]
    S35 --> Return([RETURN])
  
```

FIG. 6 is a flowchart illustrating the process of calculating max_sum_sup . The process starts with a start node labeled "max_sum_sup算出処理2START". It then proceeds to step S31: "直近のIピクチャの発生符号量を検出する" (Detect the occurrence code amount of the latest I-picture). Step S32: "max_sum_sup = VBVバッファサイズ - Iピクチャ発生量とする" (Set max_sum_sup to VBV buffer size minus I-picture occurrence amount). Step S33: "シーンチェンジであるか?" (Is it a scene change?). If the answer is "No", the process loops back to S33. If the answer is "Yes", it proceeds to step S34: "シーンチェンジのIピクチャ、および1つ前のIピクチャの符号化難易度を取得する" (Obtain the coding difficulty of the I-picture at the scene change and the I-picture immediately before it). Step S35: "2つのIピクチャの符号化難易度の差を算出し、算出されたmax_sum_supの値を、符号化難易度の変化を基に増減する" (Calculate the difference in coding difficulty of the two I-pictures, and increase or decrease the calculated max_sum_sup value based on the change in coding difficulty). The process then proceeds to a return node labeled "RETURN".

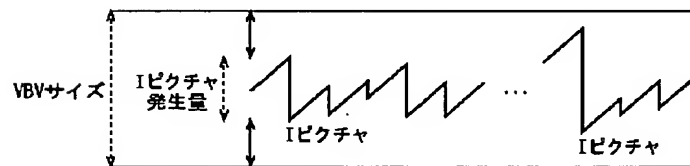
【図4】

図4



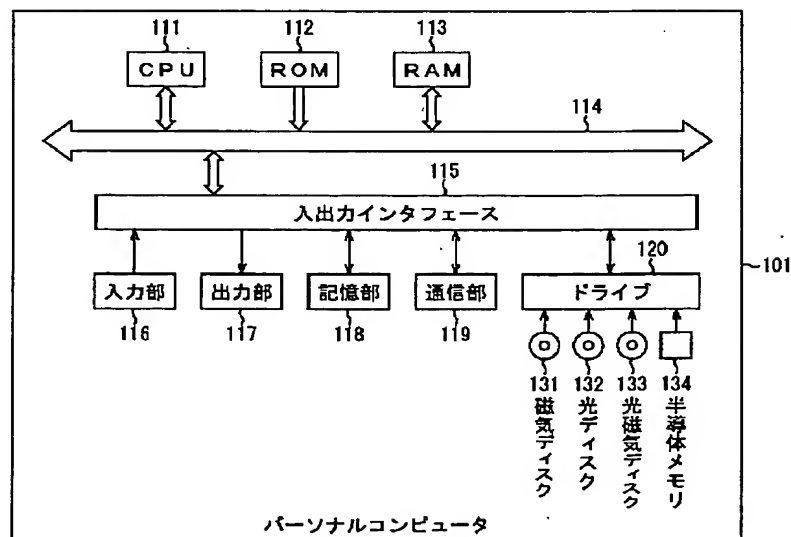
【図5】

図5



【図7】

図7



フロントページの続き

Fターム(参考) 5C059 KK01 KK22 KK35 MA00 MA05
MA14 MA23 MC11 MC38 ME01
NN01 NN21 NN43 PP05 PP06
PP07 SS20 TA46 TA60 TB03
TB04 TC02 TC03 TC10 TC14
TC16 TC20 TC27 TC38 TC41
TD03 TD05 TD06 TD12 UA02
UA32 UA33 UA38 UA39